

# Welcome to my nightmare

*or, From thick clients to web  
apps and back again*



Code Con

18 September 2004, Caves Beach, NSW



Andrew Frederick Cowie

Operational Dynamics  
Consulting Pty Ltd

I

# Problem space

Managing massive changes & upgrades to mission critical systems: what can go wrong, and why preparing for them needs to be done with exactitude



# Massive changes & upgrades

## Life cycle events

- software upgrades
- launches of new versions of websites
- database administration
- deploying security patches
- hardware refresh



# Massive changes & upgrades

## Characteristics

- complex
- numerous interdependent systems
- include people internal and external
- impact critical services organization depends on



# Massive changes & upgrades

## Characteristics

- Can only disrupt services minimally (if at all)
- Aren't allowed to go wrong
  - require considerable advance planning
  - necessary to clearly document

(See [Surviving Change](#) paper)



Problem space

Organizational Blueprints



Process



Procedures

Procedures are for these *specific events*.



# Style

## II.Begin: Site Down

### 5. Down Services

- Richard
  - a. Stop all VMs less IVRs, using `svc -d wish-*`
  - b. Stop Veritas on `oracle2`, then `oracle1`
  - c. Ensure RAID filesystems still mounted on `oracle1`
- Sarah
  - a. Stop Apache on `web1`, `web2`

### 6. Verification

- Garrett
  - a. Verify VMs down (less IVR VMs) with `svstat wish-*`
  - b. ...



# Style

## II. Section

### 5. Sequence

- Name
  - a. Step
  - b. Step
  - c. Step
- Name
  - a. Step

### 6. Sequence

- Name
  - a. Step
  - b. ...



II

# Technical architecture

Just when you thought you were safe...



# Back end

Need to be able to use version control. In our world, that means line oriented text based RCS like CVS, Subversion, or Arch

- ✗ Microsoft Word / OpenOffice Writer inappropriate
- ✗ Plain text insufficient
- ✓ HTML
- ✓ XML (use XSLT to render)



# Back end



# Presentation

## XML – (via XSLT) –> HTML

- ✓ Very flexible technology, ideal for bridging between structure and presentation. *but*
- ✗ Once you've converted to HTML, the knowledge of *which node* a generated element came from is lost.
- ✗ How do you generate URLs to actions to get information back to server?  
(XPath no help here – see next slide)



# Style

## Fully qualified names

Section III, Sequence 2, *Name*'s, Step d

- Raises the temptation of using XPath, but frankly, XPath is about selecting groups of nodes that fit a criteria, rather than specifying *specific* nodes. (You can do it, of course, but it's clumsy)



# Interactivity

## Common approach: web app

- Round trip of information relies on some unique id (typically `uid` column in database)
- embedded in each generated item's `<A href>` as a HTTP query string
- difficult to push state changes from server to client (client requests, server responds – and clients don't poll)



# Back end

## So what *do* we need to store?

- The procedure:
  - *structured* information
- That events happen:
  - *sequence* information
- XML is all about structure. Where do we put the sequence information?



# Back end

## Relational databases (SQL) no good

- x Real burden to deploy / install / maintain. (database connectivity, schema installation, don't even talk about upgrades)
- x XML doesn't really play well in rows & columns databases
- x version control means straight up XML text files really would be better



# Back end

## What back end data store?

- Our sequence data is a very simple table, no need for RDBMS power – but still require concurrency, transactions, and availability.
- How about an embedded database?
- ✓ No deployment issues (in server binary)
- Choice: Berkeley DB



# Back end

## Merge Structure and Sequence

### ✓ Structure:

- XML human readable on disk
- Load it in
- Choice: Do `id` assignment *at runtime*
- Now have `id` token to run out to clients!

### ✓ Sequence:

- simple table, in Berkeley DB
- event comes in from client, record `i++`



# Development Technology



# Development technology

## Which Platform?

- ✓ Linux/Unix servers will be present in any environment I'm likely to get business.
- ✗ But many Windows desktops. I have no ability to develop & test for that platform.
- ✓ Linux among techies (especially laptops) not uncommon, though. And its mostly sysadmins, developers, and operations managers who will be using this thing!



# Development technology

## Which Platform?

- It would be nice if it ran on Windows someday, but whatever.  
*(that's what Open Source is for)*
- ✓ And one can always make a custom bootable Linux LiveCD... so why not?
- ✓ **Choice: Linux**



# Development Technology

## Which language?

- ✗ C APIs really horrible (too low level). Don't speak C++ (not much better in any case).
- ✗ Considered Mono, but DE and debugger keep crashing. It'll be a good option in a few years.
- ✓ I'm a Java guy. Been using it (and have done client and server side work) for 7+ years. **Choice: Java**



# Development Technology

## Which widget set?

- ✗ Java **AWT** is terrible. Java **Swing** is even worse. The usual reason to use them is cross platform – not an issue here.
- **SWT** (the widget infrastructure under Eclipse) is tempting, but ditto. And, no rapid interface designer.
- ✓ Frankly, I like **GTK** and just want to write as a native **GNOME** app. And **glade** is so cool. **Choice: java-gnome bindings**



- ✗ Running client apps in Java VMs a pain in any OS. Even getting Java installed can be tricky.
- ✗ Debian Java unfriendly. But:
  - ✓ The `gcc` people have been working on a Java compiler, `gcj`. It can create class files, but more interestingly, it can instead create object code, and link it with a small run time GC, to **create native Linux executables!**
  - ✓ So far so good: programs written using `java-gnome` and `BerkeleyDB` build and run under `gcj` no problem. Run fast, too!



# Client side presentation



# Presentation

## Do the translation client side?

- Having accepted the idea of a thick client, suddenly many new options present themselves
- ✓ Could do a *second* translation client side to embed additional information which could power the GUI



# Presentation

## Spacial Model

- x Typically hierarchical structures (especially XML) are presented in tree views. But for the sequence stuff, they aren't a good UI
- ✓ **Nautilus's** spacial model presents an intriguing idea: have various layers of the procedure open in separate windows, with a `GtkListView` type navigator to step through sequential steps. [Yes, I need a picture here]



# Presentation

## What actual display tool?

- The original idea was to use `gtkhtml` to render the output of the client side XSL transform.
- If we're using some complex GTK widget display API, Do we still use XSLT?



# Presentation

## Bonus time

- ✓ Regardless, it turns out we could still do an XSLT in a web server somewhere to create a read-only, non-interactive web version for people without the client!

...which means the choice of platform is even less of a worry!



# Client-Server communication



# Intercommunication

## Not much data...

- Initial setup, just transmit the XML file (augmented with `id` assignments)
- ✓ only need to transmit minimal bit of data (`id`) for a state change event.
- future requirement – handle case of someone making a change to the procedure *while the event is running*. Presumably something to transmit frags of XML would be handy...



# Intercommunication

...so don't want to overcomplicate...

- \* I've written two way socket layer client server protocols before. Ick.
- \* Full blown **J2EE** is overkill (not to mention deployment/installation nightmare)
- **ICE** (Internet Communications Engine, a next generation CORBA) from ZeroC is tempting. Good scalability, features, robustness. Very complex API, though.



# Intercommunication

...it's just about message passing, yo!

- x XML-RPC would be terrific, (nice simple API, especially in Java) but it's HTTP
- x Java Message System (say, OpenJMS) API complex, same underlying container requirements as J2EE (ie, stands on top of a full RDBMS. Ick)
- ✓ Jabber could work! From inception, meant as a general messaging tool. Instant Messaging is just a case of that!



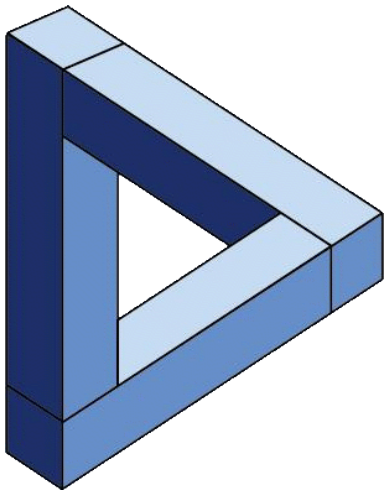
R&D continues...



Prepared on a Gentoo Linux system

running GNOME 2.6

using OpenOffice-ximian 1.1.61

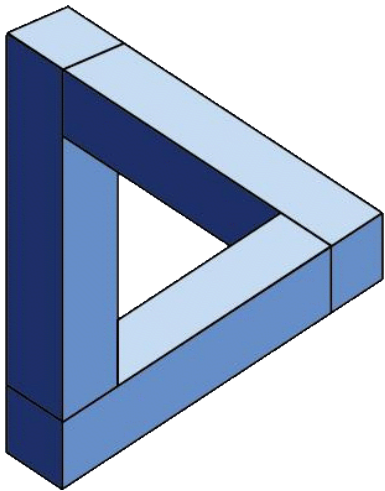


# Operational Dynamics

Sydney, Toronto, New York, London

These slides available at

[operationaldynamics.com/talks](http://operationaldynamics.com/talks)



**Andrew Frederick Cowie**

Managing Director

[andrew@operationaldynamics.com](mailto:andrew@operationaldynamics.com)

**Operational Dynamics**

Sydney, Toronto, New York, London